用户

Astra Automation

NetApp
November 17, 2026

This PDF was generated from https://docs.netapp.com/zh-cn/astra-automation-2310/reference/user_endpoints.html on November 17, 2026. Always check docs.netapp.com for the latest.

目录

用户 1
Astra API (ACC) 用户端点 1
为群组创建新用户 1
HTTP 方法和端点 1
使用说明 1
参数 1
请求正文 1
响应 4
响应 9
响应 10
响应 10
响应 11
错误 11
定义 12
列出组的用户 14
HTTP 方法和端点 14
使用说明 14
参数 14
响应 16
响应 17
响应 17
响应 18
错误 18
定义 19
检索组的用户 25
HTTP 方法和端点 25
使用说明 25
参数 25
响应 25
响应 30
响应 31
响应 31
错误 32
定义 32
修改组的用户 34
HTTP 方法和端点 34
使用说明 34
参数 34
请求正文 35

响应 . . . 39
响应 . . . 39
响应 . . . 39
响应 . . . 40
响应 . . . 40
错误 . . . 41
定义 . . . 42
为组删除用户 . . . 43
HTTP 方法和端点 . . . 44
使用说明 . . . 44
参数 . . . 44
响应 . . . 44
响应 . . . 44
响应 . . . 45
响应 . . . 45
错误 . . . 46
定义 . . . 46
创建新用户 . . . 46
HTTP 方法和端点 . . . 47
使用说明 . . . 47
参数 . . . 47
请求正文 . . . 47
响应 . . . 50
响应 . . . 55
响应 . . . 56
响应 . . . 56
响应 . . . 57
错误 . . . 57
定义 . . . 58
列出所有用户 . . . 60
HTTP 方法和端点 . . . 60
使用说明 . . . 60
参数 . . . 60
响应 . . . 62
响应 . . . 63
响应 . . . 63
响应 . . . 64
错误 . . . 64
定义 . . . 65
检索用户 . . . 71
HTTP 方法和端点 . . . 71

使用说明 71
参数 71
响应 71
响应 76
响应 77
响应 77
错误 78
定义 78
修改用户 80
HTTP 方法和端点 80
使用说明 80
参数 80
请求正文 81
响应 85
响应 85
响应 85
响应 86
响应 86
错误 87
定义 88
删除用户 89
HTTP 方法和端点 90
使用说明 90
参数 90
响应 90
响应 90
响应 91
响应 91
错误 92
定义 92

用户

Astra API (ACC) 用户端点

管理可以与 Astra Control 交互的用户。

User API 表示能够与系统交互的用户。

为群组创建新用户

为组创建新用户。使用此功能可以有效地管理组成员和权限。

HTTP 方法和端点

POST `/accounts/{account_id}/core/v1/groups/{group_id}/users`

使用说明

指示要创建的 User API 资源的所需值。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
group_id	string	path	True	包含的组资源的 ID

请求正文

指示要创建的 User API 资源的所需值。

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"

名称	Type	Required	Description
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。
firstName	string	False	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击

名称	Type	Required	Description
lastName	string	False	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	False	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
元数据	"type_astra_metadata_update"	False	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 请求示例

```
{
  "type": "application/astra-user",
  "version": "1.2",
  "firstName": "John",
  "lastName": "Doe",
  "email": "jdoe@example.com"
}
```

响应

```
Status: 201, Returns the newly created user resource in the JSON response
body.
```

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	True	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	True	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	True	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	True	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	True	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	True	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	True	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 响应示例

```
{
```

```
  "type": "application/astra-user",
  "version": "1.2",
  "id": "5bad8e04-e2fd-5c43-98d7-300580993f49",
  "state": "pending",
  "isEnabled": "true",
  "authID": "HteLK3hk28sdbDLKHReB",
  "authProvider": "cloud-central",
  "firstName": "John",
  "lastName": "Doe",
  "email": "jdoe@example.com",
  "sendWelcomeEmail": "true",
  "lastActTimestamp": "2020-08-05T14:24:54.128427Z",
  "metadata": {
    "labels": [],
    "creationTimestamp": "2022-10-06T20:58:16.305662Z",
    "modificationTimestamp": "2022-10-06T20:58:16.305662Z",
    "createdBy": "8f84cf09-8036-51e4-b579-bd30cb07b269"
  }
}
```


响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```

响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 409, Conflict
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/10",
  "title": "JSON resource conflict",
  "detail": "The request body JSON contains a field that conflicts with
```

```
an idempotent value.",
  "status": "409"
}
```

响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```

错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例

```
{
  "type": "https://astra.netapp.io/problems/2",
```

```
  "title": "Collection not found",
  "detail": "The collection specified in the request URI wasn't found.",
  "status": "404"
}
```

定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63

名称	Type	Required	Description
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

invalidFields

名称	Type	Required	Description
name	string	True	无效请求正文字段的名称
原因	string	True	请求正文字段无效的原因

列出组的用户

检索组的用户，提供集合中所有匹配资源的 JSON 数组。

HTTP 方法和端点

GET `/accounts/{account_id}/core/v1/groups/{group_id}/users`

使用说明

返回集合中所有匹配资源的 JSON 数组。如果在请求中指定了单个字段，则项目 JSON 数组将包含按指定顺序为每个匹配资源请求的字段。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
group_id	string	path	True	包含的组资源的 ID
包括	string	查询	False	指示列出集合时应返回哪些字段。 URL 示例：“include=id”、“include=id,name” OpenAPI 示例：“id” • 示例：id、email
limit	string	查询	False	指示列出集合时要返回的最大资源数。 URL 示例：“limit=2” OpenAPI 示例：“2”

名称	Type	在	Required	Description
筛选器	string	查询	False	指示在列出集合时应匹配哪些字段以返回资源。 支持的运算符有： • eq - 相等 • lt/gt - 小于/大于 • lte/gte - 小于或等于/大于或等于 URL 示例：“filter=field%20eq%20%27value%27” OpenAPI 示例：“field eq 'value'”
订单	string	查询	False	指示列出集合时资源的排序顺序。 URL 示例：“orderBy=name”，“orderBy=name%20desc” OpenAPI 示例：“name desc”
跳过	string	查询	False	指示列出集合时要跳过的资源数。 URL 示例：“skip=1” OpenAPI 示例：“1”
count	string	查询	False	指示是否应在元数据对象中返回资源总数。 URL 示例：“count=true” OpenAPI 示例：“true”

名称	Type	在	Required	Description
继续		查询	False	指示要从中恢复列表的令牌。 URL 示例： "continue=<continue token="" from="" collection="" metadata="">"</continue> OpenAPI 示例： "<continue token="" from="" collection="" metadata=""></continue>+</continue>

响应

```
Status: 200, Returns a JSON array of all matching resources in the
collection. If individual fields are specified in the request, the items
JSON array will contain the fields requested for each matching resource in
the order specified.
```

名称	Type	Required	Description
个项	数组[任意以下之一： "user_1.2_get_response_body"，数组[]]	True	
元数据	"type_astra_metadata_list"	True	

▼ 响应示例


```
{
  "type": "application/astra-users",
  "version": "1.2",
  "items": [
    [
      "11c0cfa1-4f27-4ef7-a21c-cb452fc5c9aa",
      "jdoe@example.com"
    ],
    [
      "e324ec3a-706d-4f57-a95d-1d88c03e7b73",
      "ssmith@example.com"
```

```
      ],
      [
        "3070c84d-129b-4017-82a8-2c26a42dd77e",
        "wjohns@example.com"
      ]
    ],
    "metadata": {
      "labels": [],
      "creationTimestamp": "2022-10-06T20:58:16.305662Z",
      "modificationTimestamp": "2022-10-06T20:58:16.305662Z",
      "createdBy": "8f84cf09-8036-51e4-b579-bd30cb07b269"
    }
}
```


响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	

名称	Type	Required	Description
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidParams	数组["invalidParams"]	False	无效查询参数列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/2",
  "title": "Collection not found",
  "detail": "The collection specified in the request URI wasn't found.",
  "status": "404"
}
```


定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63

名称	Type	Required	Description
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	

名称	Type	Required	Description
modifiedBy	string	False	

user_1.2_get_response_body

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	True	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	True	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	True	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	True	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	True	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	True	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	True	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

type_astra_metadata_list

名称	Type	Required	Description
继续	string	False	
count	数字	False	
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

检索组的用户

检索组的用户，提供对存储的用户 API 资源详细信息的访问权限。

HTTP 方法和端点

GET `/accounts/{account_id}/core/v1/groups/{group_id}/users/{user_id}`

使用说明

返回 JSON 响应正文中存储的 User API 资源。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
group_id	string	path	True	包含的组资源的 ID
用户 ID	string	path	True	要列出的用户集合的 ID

响应

```
Status: 200, Returns the stored User API resource in the JSON response
```

```
body.
```

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	True	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	True	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	True	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	True	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	True	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	True	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	True	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 响应示例

```
{
```

```
  "type": "application/astra-user",
  "version": "1.2",
  "id": "5bad8e04-e2fd-5c43-98d7-300580993f49",
  "state": "active",
  "isEnabled": "true",
  "authID": "HteLK3hk28sdbDLKHReB",
  "authProvider": "cloud-central",
  "firstName": "John",
  "lastName": "Doe",
  "companyName": "NetApp, Inc.",
  "email": "jdoe@example.com",
  "phone": "408-555-22222",
  "postalAddress": {
    "addressCountry": "US",
    "addressLocality": "Sunnyvale",
    "addressRegion": "California",
    "postalCode": "94089",
    "streetAddress1": "495 East Java Drive"
  },
  "sendWelcomeEmail": "true",
  "enableTimestamp": "2020-08-05T14:24:54.128427Z",
  "lastActTimestamp": "2020-08-05T14:24:54.128427Z",
  "metadata": {
    "labels": [],
    "creationTimestamp": "2022-10-06T20:58:16.305662Z",
    "modificationTimestamp": "2022-10-06T20:58:16.305662Z",
    "createdBy": "8f84cf09-8036-51e4-b579-bd30cb07b269"
  }
}
```


响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidParams	数组["invalidParams"]	False	无效查询参数列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	

名称	Type	Required	Description
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```

错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例

```
{
  "type": "https://astra.netapp.io/problems/2",
  "title": "Collection not found",
  "detail": "The collection specified in the request URI wasn't found.",
  "status": "404"
}
```

定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

修改组的用户

通过用提供的 JSON 请求正文替换整个存储资源来修改组的用户。使用此端点可以更新用户详细信息，同时保留不可修改的键值。

HTTP 方法和端点

PUT `/accounts/{account_id}/core/v1/groups/{group_id}/users/{user_id}`

使用说明

将整个存储资源替换为提供的 JSON 请求正文，同时保留不可由用户修改的键值。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
group_id	string	path	True	包含的组资源的 ID
用户 ID	string	path	True	要列出的用户集合的 ID

请求正文

将整个存储资源替换为提供的 JSON 请求正文，同时保留不可由用户修改的键值。

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	False	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	False	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	False	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	False	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	False	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	False	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	False	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	False	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 请求示例

```
{
```

```
  "type": "application/astra-user",
  "version": "1.2",
  "firstName": "John",
  "lastName": "Dale",
  "email": "jdale@example.com"
}
```


响应

```
Status: 204, Indicates if the user resource was updated.
```

响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	

名称	Type	Required	Description
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 409, Conflict
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/10",
  "title": "JSON resource conflict",
  "detail": "The request body JSON contains a field that conflicts with
an idempotent value.",
  "status": "409"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/1",
  "title": "Resource not found",
  "detail": "The resource specified in the request URI wasn't found.",
  "status": "404"
}
```

定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	**Type**	**Required**	**Description**
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63

名称	Type	Required	Description
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

invalidFields

名称	Type	Required	Description
name	string	True	无效请求正文字段的名称
原因	string	True	请求正文字段无效的原因

为组删除用户

删除组的用户，删除指定的用户 API 资源和相关资源，如通知标记、令牌和角色绑定。

HTTP 方法和端点

DELETE `/accounts/{account_id}/core/v1/groups/{group_id}/users/{user_id}`

使用说明

删除指定的用户 API 资源。

副作用

- 将删除与该用户关联的以下资源：notificationMarks、令牌以及所有引用该用户的 roleBindings。
- 对于 "authProvider" 为 "local" 的用户，"keyType" 为 "passwordHash" 的链接凭据也将被删除。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
group_id	string	path	True	包含的组资源的 ID
用户 ID	string	path	True	要列出的用户集合的 ID

响应

```
Status: 204, Indicates if the user resource was deleted.
```

响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
```

```
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidParams	数组["invalidParams"]	False	无效查询参数列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误


```
Status: 404, Not found
```


名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/1",
  "title": "Resource not found",
  "detail": "The resource specified in the request URI wasn't found.",
  "status": "404"
}
```


定义

▼ 查看定义

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

创建新用户

在 Astra API 中创建新用户，指定用户 API 资源的所需值。使用此选项可以管理 Astra 环境中的用户访问和权限。

HTTP 方法和端点

POST `/accounts/{account_id}/core/v1/users`

使用说明

指示要创建的 User API 资源的所需值。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}

请求正文

指示要创建的 User API 资源的所需值。

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。

名称	Type	Required	Description
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。
firstName	string	False	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	False	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击

名称	Type	Required	Description
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。
sendWelcomeEmail	string	False	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。

名称	Type	Required	Description
元数据	"type_astra_metadata_update"	False	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 请求示例

```
{
  "type": "application/astra-user",
  "version": "1.2",
  "firstName": "John",
  "lastName": "Doe",
  "email": "jdoe@example.com"
}
```

响应

```
Status: 201, Returns the newly created user resource in the JSON response body.
```

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"

名称	Type	Required	Description
id	string	True	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	True	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。
isEnabled	string	True	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。

名称	Type	Required	Description
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。
firstName	string	True	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击

名称	Type	Required	Description
lastName	string	True	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	True	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	True	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 响应示例

```
{
```

```
  "type": "application/astra-user",
  "version": "1.2",
  "id": "5bad8e04-e2fd-5c43-98d7-300580993f49",
  "state": "pending",
  "isEnabled": "true",
  "authID": "HteLK3hk28sdbDLKHReB",
  "authProvider": "cloud-central",
  "firstName": "John",
  "lastName": "Doe",
  "email": "jdoe@example.com",
  "sendWelcomeEmail": "true",
  "lastActTimestamp": "2020-08-05T14:24:54.128427Z",
  "metadata": {
    "labels": [],
    "creationTimestamp": "2022-10-06T20:58:16.305662Z",
    "modificationTimestamp": "2022-10-06T20:58:16.305662Z",
    "createdBy": "8f84cf09-8036-51e4-b579-bd30cb07b269"
  }
}
```


响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```

响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 409, Conflict
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/10",
  "title": "JSON resource conflict",
  "detail": "The request body JSON contains a field that conflicts with
```

```
an idempotent value.",
  "status": "409"
}
```

响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```

错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例

```
{
  "type": "https://astra.netapp.io/problems/2",
```

```
  "title": "Collection not found",
  "detail": "The collection specified in the request URI wasn't found.",
  "status": "404"
}
```

定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63

名称	Type	Required	Description
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

invalidFields

名称	Type	Required	Description
name	string	True	无效请求正文字段的名称
原因	string	True	请求正文字段无效的原因

列出所有用户

检索系统中所有用户的列表，提供对用户信息和配置的访问权限。使用此功能可有效查看或管理用户帐户。

HTTP 方法和端点

GET `/accounts/{account_id}/core/v1/users`

使用说明

返回集合中所有匹配资源的 JSON 数组。如果在请求中指定了单个字段，则项目 JSON 数组将包含按指定顺序为每个匹配资源请求的字段。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
包括	string	查询	False	指示列出集合时应返回哪些字段。 URL 示例："include=id"、"include=id,name" OpenAPI 示例："id" • 示例：id、email
limit	string	查询	False	指示列出集合时要返回的最大资源数。 URL 示例："limit=2" OpenAPI 示例："2"

名称	Type	在	Required	Description
筛选器	string	查询	False	指示在列出集合时应匹配哪些字段以返回资源。 支持的运算符有： • eq - 相等 • lt/gt - 小于/大于 • lte/gte - 小于或等于/大于或等于 URL 示例：“filter=field%20eq%20%27value%27” OpenAPI 示例：“field eq 'value'”
订单	string	查询	False	指示列出集合时资源的排序顺序。 URL 示例：“orderBy=name”，“orderBy=name%20desc” OpenAPI 示例：“name desc”
跳过	string	查询	False	指示列出集合时要跳过的资源数。 URL 示例：“skip=1” OpenAPI 示例：“1”
count	string	查询	False	指示是否应在元数据对象中返回资源总数。 URL 示例：“count=true” OpenAPI 示例：“true”

名称	Type	在	Required	Description
继续		查询	False	指示要从中恢复列表的令牌。 URL 示例： "continue=<continue token="" from="" collection="" metadata="">"</continue> OpenAPI 示例： "<continue token="" from="" collection="" metadata=""></continue>+</continue>

响应

```
Status: 200, Returns a JSON array of all matching resources in the
collection. If individual fields are specified in the request, the items
JSON array will contain the fields requested for each matching resource in
the order specified.
```

名称	Type	Required	Description
个项	数组[任意以下之一： "user_1.2_get_response_body"，数组[]]	True	
元数据	"type_astra_metadata_list"	True	

▼ 响应示例


```
{
  "type": "application/astra-users",
  "version": "1.2",
  "items": [
    [
      "11c0cfa1-4f27-4ef7-a21c-cb452fc5c9aa",
      "jdoe@example.com"
    ],
    [
      "e324ec3a-706d-4f57-a95d-1d88c03e7b73",
      "ssmith@example.com"
```

```
      ],
      [
        "3070c84d-129b-4017-82a8-2c26a42dd77e",
        "wjohns@example.com"
      ]
    ],
    "metadata": {
      "labels": [],
      "creationTimestamp": "2022-10-06T20:58:16.305662Z",
      "modificationTimestamp": "2022-10-06T20:58:16.305662Z",
      "createdBy": "8f84cf09-8036-51e4-b579-bd30cb07b269"
    }
}
```


响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	

名称	Type	Required	Description
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidParams	数组["invalidParams"]	False	无效查询参数列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/2",
  "title": "Collection not found",
  "detail": "The collection specified in the request URI wasn't found.",
  "status": "404"
}
```


定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63

名称	Type	Required	Description
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	

名称	Type	Required	Description
modifiedBy	string	False	

user_1.2_get_response_body

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	True	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	True	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	True	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	True	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	True	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	True	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	True	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

type_astra_metadata_list

名称	Type	Required	Description
继续	string	False	
count	数字	False	
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

检索用户

检索用户，在 JSON 响应正文中返回存储的用户 API 资源。使用此选项可访问用户详细信息，以便执行管理和集成任务。

HTTP 方法和端点

GET `/accounts/{account_id}/core/v1/users/{user_id}`

使用说明

返回 JSON 响应正文中存储的 User API 资源。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
用户 ID	string	path	True	要列出的用户集合的 ID

响应

```
Status: 200, Returns the stored User API resource in the JSON response
```

```
body.
```

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	True	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	True	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	True	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	True	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	True	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	True	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	True	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	True	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 响应示例

```
{
```

```
  "type": "application/astra-user",
  "version": "1.2",
  "id": "5bad8e04-e2fd-5c43-98d7-300580993f49",
  "state": "active",
  "isEnabled": "true",
  "authID": "HteLK3hk28sdbDLKHReB",
  "authProvider": "cloud-central",
  "firstName": "John",
  "lastName": "Doe",
  "companyName": "NetApp, Inc.",
  "email": "jdoe@example.com",
  "phone": "408-555-22222",
  "postalAddress": {
    "addressCountry": "US",
    "addressLocality": "Sunnyvale",
    "addressRegion": "California",
    "postalCode": "94089",
    "streetAddress1": "495 East Java Drive"
  },
  "sendWelcomeEmail": "true",
  "enableTimestamp": "2020-08-05T14:24:54.128427Z",
  "lastActTimestamp": "2020-08-05T14:24:54.128427Z",
  "metadata": {
    "labels": [],
    "creationTimestamp": "2022-10-06T20:58:16.305662Z",
    "modificationTimestamp": "2022-10-06T20:58:16.305662Z",
    "createdBy": "8f84cf09-8036-51e4-b579-bd30cb07b269"
  }
}
```


响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidParams	数组["invalidParams"]	False	无效查询参数列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	

名称	Type	Required	Description
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/2",
  "title": "Collection not found",
  "detail": "The collection specified in the request URI wasn't found.",
  "status": "404"
}
```


定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

修改用户

通过用提供的 JSON 请求正文替换整个存储资源来修改用户，同时保留不可由用户修改的键值。使用此端点可以高效地更新用户详细信息，而不会影响不可变属性。

HTTP 方法和端点

PUT `/accounts/{account_id}/core/v1/users/{user_id}`

使用说明

将整个存储资源替换为提供的 JSON 请求正文，同时保留不可由用户修改的键值。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
用户 ID	string	path	True	要列出的用户集合的 ID

请求正文

将整个存储资源替换为提供的 JSON 请求正文，同时保留不可由用户修改的键值。

名称	Type	Required	Description
type	string	True	资源的媒体类型。定义的值为： • "application/astra-user"
version	string	True	资源的版本。定义的值为： • "1.0" • "1.1" • "1.2"
id	string	False	资源的全局唯一标识符。定义的值为： • 符合 UUIDv4 架构
状态	string	False	包含指示用户操作状态的值的 JSON 字符串。定义的值为： • "pending" - 正在创建用户帐户。用户只能读取和修改用户资源（默认值）。 • 此状态对于 "authProvider" 为 "local" 的用户无效 • "active" - 用户帐户是可操作的（如果帐户设置"isEnabled" 为"true"，则用户可以登录）。 • 这是 "authProvider" 为 "local" 的用户的默认值 • "suspended" - 用户帐户已暂停（无论帐户 "isEnabled" 设置如何，用户都无法登录）。

名称	Type	Required	Description
isEnabled	string	False	指示是否允许用户登录的 JSON 字符串。定义的值为： • "true" - 允许用户登录；创建时为默认值。 • "false" - 不允许此用户登录。
authID	string	False	包含与外部标识提供程序关联的标识符的 JSON 字符串。此字符串的内容取决于指定的 authProvider。对于 "authProvider" 为 "local" 的用户，"authID" 设置为 "email" 的值。对于 "authProvider" 为 "ldap" 的用户，"authID" 设置为其来自 LDAP 服务器的可分辨名称 (DN)。
authProvider	string	False	表示 idAuth 标识符上下文的 JSON 字符串。定义的值为： • "cloud-central" - 用户身份验证来自 NetApp Cloud Central 服务 • "local" - 用户身份验证来自本地用户/密码存储 • "ldap" - 用户身份验证来自外部 LDAP 服务器。在 Astra Control Center 中，所有用户都必须具有 "local" 或 "ldap" 值。如果未指定值，则选择默认值 "local"。在 Astra Control Service 中，不支持将 "authProvider" 设为 "local" 或 "ldap"。如果未指定值，则默认为 "cloud-central"。

名称	Type	Required	Description
firstName	string	False	包含用户指定的长格式名字的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
lastName	string	False	包含用户指定的长格式姓氏的 JSON 字符串。定义的值为： • 最小长度：0 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
companyName	string	False	包含与用户关联的公司名称的 JSON 字符串。如果此密钥不存在，则表示未提供公司名称。定义的值是： • 最小长度：1 • 最大长度：63 • 此字符串经过内部验证，可防止 XSS 、Unicode、目录遍历和 SQL 注入攻击
电子邮件	string	False	包含用户指定的电子邮件地址的 JSON 字符串。
电话	string	False	包含用户电话号码的 JSON 字符串。如果此密钥不存在，则表示未提供电话号码。
postalAddress	"postalAddress"	False	JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	Type	Required	Description
sendWelcomeEmail	string	False	表示是否在验证后发送欢迎电子邮件的 JSON 字符串。如果在创建时未指定，则 ACS 的值将设置为 "true"，Astra Control Center 的值将设置为 "false"。定义的值为： • "true" - 将发送一封欢迎电子邮件；创建时为默认值 • "false" - 不会发送欢迎电子邮件。对于 "authProvider" 为 "local" 或 "ldap" 的用户，此值将被忽略并设置为 "false"。
enableTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次启用用户的时间。定义的值为： • 符合 ISO-8601 日期时间方案
lastActTimestamp	string	False	包含 ISO-8601 格式时间戳的 JSON 字符串，指示上次代表此用户调用身份 API 的时间。（注：这是昂贵的。）定义的值为： • 符合 ISO-8601 日期时间方案
元数据	"type_astra_metadata_update"	False	与资源关联的客户端和服务指定的元数据。定义的值为： • 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

▼ 请求示例

```
{
```

```
  "type": "application/astra-user",
  "version": "1.2",
  "firstName": "John",
  "lastName": "Dale",
  "email": "jdale@example.com"
}
```

响应

```
Status: 204, Indicates if the user resource was updated.
```

响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/3",
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```

响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	

名称	Type	Required	Description
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 409, Conflict
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidFields	数组["invalidFields"]	False	无效请求正文字段列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/10",
  "title": "JSON resource conflict",
  "detail": "The request body JSON contains a field that conflicts with
an idempotent value.",
  "status": "409"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/1",
  "title": "Resource not found",
  "detail": "The resource specified in the request URI wasn't found.",
  "status": "404"
}
```

定义

▼ 查看定义

postalAddress

JSON 对象包含指定地址的字段集合。如果此 key 不存在，则表示未提供地址。如果此 key 存在，则所有六个包含的字段都将存在。如果用户未提供任何内容，则字符串将为空。

名称	**Type**	**Required**	**Description**
addressCountry	string	True	包含与邮政地址关联的国家/地区名称或国家/地区代码的 JSON 字符串，如 ISO 3166 Alpha-2 中所指定。定义的值是： • 最小长度：2 • 最大长度：2
addressLocality	string	True	包含与邮政地址关联的地点（例如城市）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
addressRegion	string	True	包含与邮政地址关联的区域（例如州）的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
postalCode	string	True	包含与邮政地址关联的邮政编码的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63
streetAddress1	string	True	包含与邮政地址关联的街道地址的 JSON 字符串。定义的值是： • 最小长度：1 • 最大长度：63

名称	Type	Required	Description
streetAddress2	string	False	包含可选辅助街道地址信息、邮政信箱等的 JSON 字符串。定义的值为： • 最小长度：1 • 最大长度：63

类型 Astra 标签

名称	Type	Required	Description
name	string	True	
value	string	True	

type_astra_metadata_update

与资源关联的客户端和服务指定的元数据。定义的值为：

- 符合 Astra 元数据架构如果在创建时未指定，则将创建没有标签的元数据对象。如果在更新时未指定，则元数据对象的标签、creationTimestamp 和 createdBy 将保留而不进行修改。

名称	Type	Required	Description
labels	数组["类型 Astra 标签"]	False	
creationTimestamp	string	False	
modificationTimestamp	string	False	
已创建	string	False	
modifiedBy	string	False	

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

invalidFields

名称	Type	Required	Description
name	string	True	无效请求正文字段的名称
原因	string	True	请求正文字段无效的原因

删除用户

从 Astra API 中删除用户，删除相关资源，例如通知标记、令牌和角色绑定。

HTTP 方法和端点

DELETE `/accounts/{account_id}/core/v1/users/{user_id}`

使用说明

删除指定的用户 API 资源。

副作用

- 将删除与该用户关联的以下资源：notificationMarks、令牌以及所有引用该用户的 roleBindings。
- 对于 "authProvider" 为 "local" 的用户，"keyType" 为 "passwordHash" 的链接凭据也将被删除。

参数

名称	Type	在	Required	Description
account_id	string	path	True	包含帐户资源的 ID • 示例：{{.Account}}
用户 ID	string	path	True	要列出的用户集合的 ID

响应

```
Status: 204, Indicates if the user resource was deleted.
```

响应

```
Status: 401, Unauthorized
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/3",
```

```
  "title": "Missing bearer token",
  "detail": "The request is missing the required bearer token.",
  "status": "401"
}
```


响应

```
Status: 400, Bad request
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	
invalidParams	数组["invalidParams"]	False	无效查询参数列表

▼ 响应示例


```
{
  "type": "https://astra.netapp.io/problems/5",
  "title": "Invalid query parameters",
  "detail": "The supplied query parameters are invalid.",
  "status": "400"
}
```


响应

```
Status: 403, Forbidden
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 响应示例

```
{
  "type": "https://astra.netapp.io/problems/11",
  "title": "Operation not permitted",
  "detail": "The requested operation isn't permitted.",
  "status": "403"
}
```


错误

```
Status: 404, Not found
```

名称	Type	Required	Description
type	string	True	
标题	string	True	
详细信息	string	True	
状态	string	True	
correlationID	string	False	

▼ 错误响应示例


```
{
  "type": "https://astra.netapp.io/problems/1",
  "title": "Resource not found",
  "detail": "The resource specified in the request URI wasn't found.",
  "status": "404"
}
```


定义

▼ 查看定义

invalidParams

名称	Type	Required	Description
name	string	True	无效查询参数的名称
原因	string	True	查询参数无效的原因

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。